Computer Science

PARE: A PRUNING DEVICE FOR AUTOMATED MULTI-DOCUMENT TEXT SUMMARIZATION

Lucas Kreger, Math and Computer Science Department, Macalester College, MN, 55105 (*lkreger@macalester.edu*)
Rachel Tornheim, Computer Science Department, Wellesley College, MA, 02431 (*rtornhei@wellesley.edu*)
Scott Thede(*), Computer Science Department, DePauw University, IN 46135 (*sthede@depauw.edu*)

Automated multi-document text summarization is the generation of a summary of a set of documents by a computer. The summary should contain the key information from the document set with minimal redundancy. Prior work in this area has focused mainly on extraction techniques, in which pertinent pieces of each document are identified, concatenated, and returned. These methods typically work within a single scope - the paragraph, the sentence, or the phrase - and return verbatim segments of the original documents. These strategies generally also have the program determine the main topics of the document, limiting user input to adjusting the size of the summary to be produced.

PARE (Pruner And Redundancy Eliminator) is a program that provides a graphical user interface (GUI) and a back end program for tri-phase multi-document pruning-based summarization. It differs from previous work on automated multi-document text summarization in several ways. First, PARE's general approach is fundamentally different from other systems. Instead of seeking out portions of text that are the most relevant to the main topics and keeping them, it locates and removes pieces of text that either are not germane to the topics or do not contribute significant content to the text, returning to the user what is left after this pruning process is complete. The second novel aspect of PARE is its use of multiple scopes. Rather than manipulating the text at only one level, it executes three independent waves of pruning at the levels of the word, the paragraph, and the sentence. Finally, PARE provides a GUI through which the user can enter keywords that guide the pruning process, as well as thresholds that determine how much text is removed by the paragraph and sentence pruners, allowing the user to control the focus, the size, and the quality of the summary.

PARE's first step operates at the level of the word. Here, it employs a series of rules that reword phrases to restate them in fewer words, and delete words and groups of words that contribute little or no content to the text. This word pruning removes only a small percentage of the text but it leaves virtually all of the information of the documents intact.

PARE's second stage of pruning occurs at the paragraph level. Using keywords supplied by the user, the program assigns each paragraph of the original document set a *pertinence score* which reflects how much that paragraph relates to the keywords. This score is incremented each time the given paragraph contains a keyword, a word with the same root as a keyword, a word that is a synonym of a keyword, or a word that shares a synonym with a keyword, assigning more weight to closer matches. Paragraphs with pertinence scores below a user-defined threshold are removed.

Finally, PARE moves to the level of the sentence for the third and final phase of pruning. Here, it finds and prunes sentences that are determined to be redundant with other sentences in the document set. It compares every sentence to every other sentence and records information matches between them based on words that the sentences have in common, and the degree of match, which reflects whether the common words are identical or merely synonymous.

Several additional features improve the sentence pruner's effectiveness. First, to avoid the matching of functional words, a list was made of words that should not be matched, such as pronouns and determiners. Second, to prevent a ten word sentence from being deemed 100% redundant by matching each of its ten words with one word from ten different sentences, the algorithm takes as a parameter a minimum number of words that must match between two sentences for the match to 'stick'. Finally, PARE ensures that at least one copy of any redundant information remains when the process is finished. To avoid double deletion of information, the source of each match is recorded, and a sentence can only be removed if the percentage of redundant sources remaining meets a user-defined threshold.

The final summary is displayed in a window within the PARE program as a series of bulleted sentences and clusters of consecutive sentences, grouped by document. Each grouping contains a hyperlink to that section of text within the context of a word- and paragraph- but not sentence-pruned document. Thus, the user can 'zoom-in,' quickly recovering additional information in context. Further clicks on the hyperlinks or on the zoom in and out buttons reveal the document set at different stages of the pruning process. This feature means that PARE can afford to be more aggressive in the summarization process, since the user can easily retrieve in context the pieces of text that have been pruned.

Finally, PARE uses a substantial amount of caching and dynamic programming to assure that if the document set does not change, the parameters for the generation of the summary can be altered (by way of GUI elements) and the new summary can be generated virtually instantaneously. Thus, PARE provides the user with a variety of easy and efficient ways to navigate a series of documents and glean the relevant general information as well as the pertinent details.

Acknowledgement: NLF REU grant number EIA-9911626 supported this work.